IN THE CLAIMS:

1. (original) A processor comprising:

an execution unit to execute a set of instructions; and

an instruction fetching mechanism that retrieves the set of instructions to be executed by the execution unit, at least one of the set of instructions comprising a single instruction that provides for execution of other instructions of the set of instructions in accordance with multiple looping constructs.

2. (original) The processor of claim 1, wherein the single instruction initializes a plurality of loops for later execution.

3. (original) The processor of claim 1, wherein the multiple looping constructs are implemented in a nested structure.

4. (original) The processor of claim 1, wherein the single instruction includes a plurality of loop termination conditions.

5. (original) The processor of claim 1, wherein the single instruction includes at least one field that identifies a location of a last instruction of a loop.

6. (original) A method of performing an instruction for use by a processor, the method comprising:

fetching the instruction from a memory source;

decoding the instruction to identify an instruction type; and

initializing a plurality of dedicated loop storage elements corresponding to a plurality of different loops to be executed using a single instruction.

7. (original) The method of claim 6, wherein the storage elements include at least one of the following: an end-of-loop indicator, a start of loop indicator, a loop count, a loop register, and a condition code selection.

8. (original) The method of claim 6, wherein the storage elements include a loop address for a start of a loop.

9. (original) A method of performing an instruction for use by a processor, the method comprising:

fetching the instruction from a memory source;

decoding the instruction to identify an instruction type; and

determining a loop type for a single instruction that is to execute a plurality of different loops, the loop type comprising one of a conditional and non-conditional type of loop termination.

10. (original) A method of executing at least one instruction by a processor, the method comprising:

fetching a first instruction from a memory source;

decoding the first instruction to identify an instruction type; and

determining a loop type selected from one of a conditional and non-conditional type of loop termination for a loop that contains more than one instruction other than the first instruction.

11. (original) A method of executing at least one instruction by a processor, the method comprising:

fetching a first instruction from a memory source;

decoding the first instruction to identify an instruction type; and

determining a loop type selected from one of a conditional and non-conditional type of loop termination for a loop that contains at least one instruction that may be interrupted during loop execution.

12. – 15. (cancel)

16. (original) A processor instruction comprising:

a first field that indicates a first termination condition for a first execution loop; and

a second field that indicates a second termination condition for a second execution loop.

17. (original) The processor instruction of claim 16, wherein the first execution loop is a same loop as the second execution loop.

18. (original) The processor instruction of claim 16, further comprising a third field that indicates a first end-of-loop location.

19. (original) The processor instruction of claim 18, further comprising a fourth field that indicates a second end-of-loop location.

20. (New) A method of performing an instruction for use by a processor, the method comprising:

fetching a single instruction from a memory source, wherein the single instruction provides for execution of multiple looping constructs, the multiple looping constructs including a first loop and a second loop to be executed;

decoding the single instruction; and

in response to decoding the single instruction, using information provided by the single instruction to initialize a plurality of loop storage elements corresponding to the first loop and the second loop.

21. (New) The method of claim 20, wherein the multiple looping constructs are nested such that the first loop is nested within the second loop.

22. (New) The method of claim 20, wherein the information provided by the single instruction to initialize the plurality of loop storage elements includes at least one termination condition for each of the first loop and the second loop.

23. (New) The method of claim 22, wherein the at least one termination condition for each of the first loop and the second loop comprises at least one of a loop count value and a condition code.

24. (New) The method of claim 22, further comprising:

performing a first set of logic relating to the at least one termination condition for the first loop; and

performing a second set of logic relating to the at least one termination condition for the second loop.

25. (New) The method of claim 20, wherein the information provided by the single instruction to initialize the plurality of loop storage elements includes an end of loop indication for each of the first loop and the second loop.

26. (New) The method of claim 20, wherein the single instruction is a first instruction of the first loop and a first instruction of the second loop, the first instruction of the first loop and the first instruction of the second loop being a same instruction at a same instruction address.

27. (New) The method of claim 26, wherein the single instruction is one of a beginning instruction of the first and second loops and an ending instruction of the first and second loop.

28. (New) The method of claim 20, further comprising:

determining a loop type corresponding to the single instruction, wherein the loop type is selected from one of a conditional and non-conditional type.